



南京航空航天大学
NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

数据库原理 上机实验报告

学 院： 计算机科学与技术学院
/人工智能学院/软件学院
专 业： 人工智能培优班
班 级： 1621001
姓 名： 陈秋宇
学 号： 162110124
指导教师： 毛宇光

实验一：SQL 定义功能、数据插入

1.建立教学数据库的三个基本表：

S(Sno,Sname,Sgender,Sage,Sdept) 学生（学号，姓名，性别，年龄，系）

SC(Sno,Cno,Grade) 选课（学号，课程号，成绩）

C(Cno,Cname,Cpno,Ccredit) 课程（课程号，课程名，先行课，学分）

（1）创建三个基本表：

```
SQL> CREATE TABLE Student
  2  (Sno CHAR(8) PRIMARY KEY,
  3   Sname CHAR(20) UNIQUE,
  4   Sgender CHAR(4),
  5   Sbirthdate DATE,
  6   Smajor CHAR(20)
  7  );
```

表已创建。

```
SQL> CREATE TABLE Course
  2  (Cno CHAR(5) PRIMARY KEY,
  3   Cname CHAR(20) UNIQUE,
  4   Ccredit SMALLINT,
  5   Cpno CHAR(5)
  6  );
```

表已创建。

```
SQL> CREATE TABLE SC
  2  (Sno CHAR(8),
  3   Cno CHAR(5),
  4   Grade SMALLINT CHECK (Grade>=0 AND Grade<=100),
  5   Semester CHAR(5),
  6   Teachingclass CHAR(8) CHECK (Teachingclass LIKE '____-__'),
  7   PRIMARY KEY(Sno,Cno),
  8   FOREIGN KEY(Sno) REFERENCES Student(Sno),
  9   FOREIGN KEY(Cno) REFERENCES Course(Cno)
 10 );
```

表已创建。

显示已创建的三个基本表：

```
SQL> DESC Student
名称                               是否为空? 类型
-----
SNO                                NOT NULL CHAR(8)
SNAME                             CHAR(20)
SGENDER                           CHAR(4)
SBIRTHDATE                        DATE
SMAJOR                            CHAR(20)

SQL> DESC Course
名称                               是否为空? 类型
-----
CNO                                NOT NULL CHAR(5)
CNAME                             CHAR(20)
CCREDIT                           NUMBER(38)
CPNO                              CHAR(5)

SQL> DESC SC
名称                               是否为空? 类型
-----
SNO                                NOT NULL CHAR(8)
CNO                                NOT NULL CHAR(5)
GRADE                             NUMBER(38)
SEMESTER                          CHAR(5)
TEACHINGCLASS                     CHAR(8)
```

2.DROP TABLE、ALTER TABLE、CREATE INDEX、DROP INDEX 及 INSERT

语句输入数据。

(1) 删除表格:

```
SQL> CREATE TABLE TEST
2 (Testtext CHAR(20));
```

表已创建。

```
SQL> DROP TABLE TEST;
```

表已删除。

(2) 修改表格:

```
SQL> ALTER TABLE Student
2 DROP COLUMN Sgender;
```

表已更改。

```
SQL> ALTER TABLE Student
2 ADD Sgender CHAR(2) CHECK(Sgender IN ('男','女'));
```

表已更改。

(3) 创建和删除索引:

```
SQL> CREATE INDEX INDEX_STUDENT
2 ON STUDENT(SBIRTHDATE);
```

索引已创建。

```
SQL> DROP INDEX INDEX_STUDENT;
```

索引已删除。

(4) 插入数据:

```
SQL> INSERT INTO STUDENT(SNO, SNAME, SGENDER, SBIRTHDATE, SMAJOR)
2 VALUES('20180001', '李勇', '男', TO_DATE('2000-03-08', 'YYYY-MM-DD'), '信息安全');
```

已创建 1 行。

其余插入语句与之类似，故略去，下面展示插入数据后的结果：

```
SQL> SELECT * FROM STUDENT;
```

SNO	SNAME	SBIRTHDATE	SMAJOR	SG
20180001	李勇	08-3月 -00	信息安全	男
20180002	刘晨	01-9月 -99	计算机科学与技术	女
20180003	王敏	01-8月 -01	计算机科学与技术	女
20180004	张立	08-1月 -00	计算机科学与技术	男
20180005	陈新奇	01-11月-01	信息管理与信息系统	男
20180006	赵明	12-6月 -00	数据科学与大数据技术	男
20180007	王佳佳	07-12月-01	数据科学与大数据技术	女

已选择7行。

```
SQL> SELECT * FROM COURSE;
```

CNO	CNAME	CCREDIT	CPNO
81001	程序设计基础与C语言	4	
81002	数据结构	4	81001
81003	数据库原理	4	81002
81004	信息系统概论	4	81003
81005	操作系统	4	81001
81006	Python语言	3	81002
81007	离散数学	4	
81008	大数据技术概论	4	81003

已选择8行。

```
SQL> SELECT * FROM SC;
```

SNO	CNO	GRADE	SEMES	TEACHING
20180001	81001	85	20192	81001-01
20180001	81002	96	20201	81002-01
20180001	81003	87	20202	81003-01
20180002	81001	80	20192	81001-02
20180002	81002	98	20201	81002-01
20180002	81003	71	20202	81003-02
20180003	81001	81	20192	81001-01
20180003	81002	76	20201	81002-02
20180004	81001	56	20192	81001-02
20180004	81003	97	20201	81002-02
20180005	81003	68	20202	81003-01

已选择11行。

实验二：数据查询

1.查询选修1号课程的学生学号与姓名。

```
SQL> SELECT STUDENT.SNO, STUDENT.SNAME
2 FROM STUDENT, SC
3 WHERE SC.CNO='81001' AND STUDENT.SNO=SC.SNO;
```

SNO	SNAME
20180001	李勇
20180002	刘晨
20180003	王敏
20180004	张立

2.查询选修课程名为数据库原理的学生学号与姓名。

```
SQL> SELECT STUDENT.SNO, STUDENT.SNAME
2 FROM STUDENT, COURSE, SC
3 WHERE COURSE.CNAME='数据库原理' AND COURSE.CNO = SC.CNO AND SC.SNO=STUDENT.SNO;
```

SNO	SNAME
20180001	李勇
20180002	刘晨
20180004	张立
20180005	陈新奇

3.查询不选1号课程的学生学号与姓名。

```
SQL> SELECT SNO, SNAME
2 FROM STUDENT
3 WHERE NOT EXISTS
4 (SELECT *
5 FROM SC
6 WHERE SC.CNO='81001' AND SC.SNO=STUDENT.SNO
7 );
```

SNO	SNAME
20180005	陈新奇
20180006	赵明
20180007	王佳佳

4.查询学习全部课程学生姓名。

```
SQL> SELECT SNAME
2 FROM STUDENT
3 WHERE NOT EXISTS
4 (SELECT *
5 FROM COURSE
6 WHERE NOT EXISTS
7 (SELECT *
8 FROM SC
9 WHERE SC.SNO=STUDENT.SNO AND SC.CNO=COURSE.CNO));
```

未选定行

5.查询所有学生除了选修1号课程外所有成绩均及格的学生的学号和平均成绩，其结果按平均成绩的降序排列。

```
SQL> SELECT SNO, AVG(GRADE)
2 FROM SC
3 HAVING NOT EXISTS
4 (SELECT *
5 FROM SC
6 WHERE CNO<>'81001' AND GRADE<60)
7 GROUP BY SNO
8 ORDER BY AVG(GRADE) DESC;
```

SNO	AVG(GRADE)
20180001	89.3333333
20180002	83
20180003	78.5
20180004	76.5
20180005	68

6. 查询选修数据库原理成绩第 2 名的学生姓名。

```
SQL> SELECT SNAME
  2 FROM (
  3 SELECT STUDENT.SNAME
  4 FROM STUDENT, COURSE, SC
  5 WHERE COURSE.CNAME='数据库原理' AND COURSE.CNO=SC.CNO AND SC.SNO
  6 ORDER BY GRADE DESC)
  7 WHERE ROWNUM<=2
  8 MINUS
  9 SELECT SNAME
 10 FROM (
 11 SELECT STUDENT.SNAME
 12 FROM STUDENT, COURSE, SC
 13 WHERE COURSE.CNAME='数据库原理' AND COURSE.CNO=SC.CNO AND SC.SNO
 14 ORDER BY GRADE DESC)
 15 WHERE ROWNUM<=1;

SNAME
-----
李勇
```

此处其实可以无需如此复杂地使用 MINUS 子句，根据课本内容可直接使用 LIMIT 子句，但是 Oracle11g 不支持 LIMIT 子句。使用 LIMIT 子句操作如下：

```
SQL> SELECT SNAME
  2 FROM (
  3 SELECT STUDENT.SNAME
  4 FROM STUDENT, COURSE, SC
  5 WHERE COURSE.CNAME='数据库原理' AND COURSE.CNO=SC.CNO AND SC.SNO
  6 ORDER BY GRADE DESC)
  7 LIMIT 2 OFFSET 1;
```

或者，可以直接使用 ROWNUM 范围控制加以控制输出的行数，但是在操作过程中依然遇到问题导致输出为空。使用 ROWNUM 子句操作如下：

```
SQL> SELECT SNAME
  2 FROM (
  3 SELECT STUDENT.SNAME
  4 FROM STUDENT, COURSE, SC
  5 WHERE COURSE.CNAME='数据库原理' AND COURSE.CNO=SC.CNO AND SC.SNO
  6 ORDER BY GRADE DESC)
  7 WHERE ROWNUM<=2 AND ROWNUM>1;
```

7.查询所有 4 个学分课程中有 3 门以上（含 3 门）课程获 80 分以上（含 80 分）的学生的姓名。

```
SQL> SELECT SNAME
  2   FROM STUDENT, COURSE, SC
  3   WHERE STUDENT.SNO=SC.SNO AND SC.CNO=COURSE.CNO AND COURSE.CCREDI
T=4 AND GRADE>=80
  4   GROUP BY SNAME
  5   HAVING COUNT(*)>=3;

SNAME
-----
李勇
```

8.查询选课门数唯一的学生的学号。

```
SQL> SELECT SNO
  2   FROM SC
  3   GROUP BY SNO
  4   HAVING COUNT(CNO)=1;

SNO
-----
20180005
```

9.SELECT 语句中各种查询条件的实验。

本题对课本中提到的查询相关子句及其功能进行汇总整理。

- (1) **SELECT**: 查询功能基本子句;
- (2) **ALL / DISTINCT**: 用于 SELECT 后, 表示给出所有/去除重复的查询结果;
- (3) **WHERE**: 用于添加表或视图中对元组进行筛选的条件表达式;
- (4) **BETWEEN ... AND ...**: 用于表达所在范围;
- (5) **IN**: 用于确定是否在某个集合中;
- (6) **IS NULL**: 用于确定空值;
- (7) **AND / OR / NOT**: 用于多重条件的逻辑运算;
- (8) **LIKE**: 用于字符串匹配;
- (9) **%**: 任意长度字符串通配符;
- (10) **_**: 单个字符通配符;
- (11) **ESCAPE**: 用于换码字符标记, 以解除通配字符限定;
- (12) **ORDER BY ... ASC / DESC**: 用于对查询结果升序/降序排列;

- (13) **COUNT**: 用于统计个数的聚集函数;
- (14) **SUM / AVG**: 用于计算求和/平均值的聚集函数;
- (15) **MAX / MIN**: 用于计算最大值/最小值的聚集函数;
- (16) **GROUP BY**: 用于按值合并分组的子句;
- (17) **HAVING**: 用于分组后的条件筛选;
- (18) **LIMIT ... OFFSET ...**: 用于控制查询结果行数;
- (19) **ROWNUM**: 用于控制查询结果行数;
- (20) **LEFT / RIGHT OUTER JOIN**: 左/右外连接子句;
- (21) **ANY / ALL / SOME**: 表达任意/所有/部分的谓词修饰符;
- (22) **EXISTS**: 表达存在量词的逻辑谓词, 且 SQL 中无全称两量词;
- (23) **UNION / INTERSECT / EXCEPT/MINUS**: 对查询结果做交集/并集/差集;
- (24) **AS**: 在派生表或视图中可能使用到的关键字。

实验三：数据修改、删除

1. 把 1 号课程的非空成绩提高 10%。

```
SQL> UPDATE SC
2 SET GRADE = 1.1 * GRADE
3 WHERE CNO = '81001' AND GRADE IS NOT NULL;
```

已更新4行。

对比数据修改前后的 GRADE 情况以检验正确性。

数据修改前:

```
SQL> SELECT *
2 FROM SC
3 WHERE CNO = '81001';
```

SNO	CNO	GRADE	SEMES	TEACHINGCLASS
20180001	81001	85	20192	81001-01
20180002	81001	80	20192	81001-02
20180003	81001	81	20192	81001-01
20180004	81001	56	20192	81001-02

数据修改后:

```
SQL> SELECT *
  2  FROM SC
  3  WHERE CNO = '81001' ;
```

SNO	CNO	GRADE	SEMES	TEACHINGCLASS
20180001	81001	94	20192	81001-01
20180002	81001	88	20192	81001-02
20180003	81001	89	20192	81001-01
20180004	81001	62	20192	81001-02

此外,为了保证成绩数据的合理性与有效性,可以在修改数据前设置一个触发器,使得成绩数值最高不超过 100。

```
SQL> CREATE OR REPLACE TRIGGER TRIGGER_GRADE_UP
  2  BEFORE UPDATE ON SC
  3  FOR EACH ROW
  4  BEGIN
  5  IF (:NEW.GRADE > 100) THEN
  6  :NEW.GRADE := 100;
  7  END IF;
  8  END;
  9  /
```

触发器已创建

2.在 SC 表中删除课程名为数据库原理的成绩的元组。

```
SQL> DELETE
  2  FROM SC
  3  WHERE CNO=(SELECT CNO
  4  FROM COURSE
  5  WHERE CNAME='数据库原理');
```

已删除4行。

对比数据删除前后的数据库原理选课情况以检验正确性。

数据删除前:

```
SQL> SELECT *
  2 FROM SC, COURSE
  3 WHERE SC.CNO=COURSE.CNO AND COURSE.CNAME='数据库原理';
```

SNO	CNO	GRADE	SEMES	TEACHINGCLASS	CNO	CNAME

	CCREDIT CPNO					

20180001	81003	87	20202	81003-01	81003	数据库原理
	4 81002					
20180002	81003	71	20202	81003-02	81003	数据库原理
	4 81002					
20180004	81003	97	20201	81002-02	81003	数据库原理
	4 81002					

SNO	CNO	GRADE	SEMES	TEACHINGCLASS	CNO	CNAME

	CCREDIT CPNO					

20180005	81003	68	20202	81003-01	81003	数据库原理
	4 81002					

数据删除后：

```
SQL> SELECT *
  2 FROM SC, COURSE
  3 WHERE SC.CNO=COURSE.CNO AND COURSE.CNAME='数据库原理';
```

未选定行

3.在 Student 和 SC 表中删除学号为 20180003 的所有数据。

```
SQL> DELETE
  2 FROM SC
  3 WHERE SNO='20180003';
```

已删除2行。

```
SQL> DELETE
  2 FROM STUDENT
  3 WHERE SNO='20180003';
```

已删除 1 行。

对比数据删除前后的 20180003 学号学生信息以检验正确性。

数据删除前：

```
SQL> SELECT *
  2 FROM STUDENT
  3 WHERE SNO='20180003';
```

SNO	SNAME	SBIRTHDATE	SDEPT	SGENDER
20180003	王敏	01-8月 -01	计算机科学与技术	女

```
SQL> SELECT *
  2 FROM SC
  3 WHERE SNO='20180003';
```

SNO	CNO	GRADE	SEMES	TEACHINGCLASS
20180003	81001	89	20192	81001-01
20180003	81002	76	20201	81002-02

数据删除后：

```
SQL> SELECT *
  2 FROM STUDENT
  3 WHERE SNO='20180003';
```

未选定行

```
SQL> SELECT *
  2 FROM SC
  3 WHERE SNO='20180003';
```

未选定行

值得注意的是，在删除时需要注意删除 SC 表数据和 Student 表数据的顺序。因为在定义表时，规定了 Student.Sno 是 SC.Sno 的外码，所以如果先删除 SC 表数据会导致违反数据完整性约束条件而删除失败。应当先删除 SC 表的子记录，再删除 Student 的相关数据。

为了保证后续实验结果显示的完整性，这里对原表数据进行恢复后再进行后续实验。

实验四：视图的操作

1. 建立男学生的视图，属性包括学号、姓名、选修课程名和成绩。

```
SQL> CREATE VIEW MALE
2 AS
3 SELECT STUDENT.SNO, STUDENT.SNAME, COURSE.CNAME, SC.GRADE
4 FROM STUDENT, COURSE, SC
5 WHERE STUDENT.SNO=SC.SNO AND COURSE.CNO=SC.CNO AND STUDENT.SGENDER='男';
```

视图已创建。

查看男学生视图情况（这里已提前对实验三中删除的所有数据进行了恢复）：

```
SQL> SELECT *
2 FROM MALE;
```

SNO	SNAME	CNAME	GRADE
20180001	李勇	数据结构	96
20180001	李勇	数据库原理	87
20180001	李勇	程序设计基础与C语言	85
20180004	张立	数据库原理	97
20180004	张立	程序设计基础与C语言	56
20180005	陈新奇	数据库原理	68

已选择6行。

2.在男学生视图中查询平均成绩大于 80 分的学生学号与姓名。

```
SQL> SELECT SNO, SNAME
2 FROM MALE
3 GROUP BY SNO, SNAME
4 HAVING AVG(GRADE)>80;
```

SNO	SNAME
20180001	李勇

在该视图查询中，尽管 GROUP BY 子句仅需 SNO 就可以实现分组，但是如果仅添加 SNO 分组依据，Oracle 会报错称“不是 GROUP BY 表达式”。这里 SQL 要求 SELECT 后面出现的属性必须全部出现在 GROUP BY 后面，反之则不然。

实验五：库函数，授权控制

1.计算每个学生有成绩的课程门数、平均成绩。

使用查询语句统计计算：

```
SQL> SELECT STUDENT.SNO, STUDENT.SNAME, COUNT(SC.CNO), AVG(SC.GRADE)
2 FROM STUDENT
3 LEFT JOIN SC ON SC.SNO=STUDENT.SNO AND SC.GRADE IS NOT NULL
4 GROUP BY STUDENT.SNO, STUDENT.SNAME
5 ORDER BY STUDENT.SNO ASC;
```

SNO	SNAME	COUNT(SC.CNO)	AVG(SC.GRADE)
20180001	李勇	3	89.3333333
20180002	刘晨	3	83
20180003	王敏	2	78.5
20180004	张立	2	76.5
20180005	陈新奇	1	68
20180006	赵明	0	
20180007	王佳佳	0	

已选择7行。

这里为了使没有成绩的学生信息也显示出来，没有使用简单的内连接语法，而是使用了左外连接方法。

使用过程统计计算：

```
SQL> CREATE OR REPLACE PROCEDURE PRINTGRADES IS
2 BEGIN
3   FOR EACH_STUDENT IN
4     (SELECT STUDENT.SNO, STUDENT.SNAME, COUNT(SC.CNO) AS CNO_COUNT, AVG(SC.GRADE) AS GRADE_AVG
5     FROM STUDENT
6     LEFT JOIN SC ON SC.SNO = STUDENT.SNO AND SC.GRADE IS NOT NULL
7     GROUP BY STUDENT.SNO, STUDENT.SNAME
8     ORDER BY STUDENT.SNO ASC)
9   LOOP
10    DBMS_OUTPUT.PUT_LINE(' 学号: ' || EACH_STUDENT.SNO || ' ');
11    DBMS_OUTPUT.PUT_LINE(' 姓名: ' || EACH_STUDENT.SNAME || ' ');
12    DBMS_OUTPUT.PUT_LINE(' 有成绩课程数: ' || EACH_STUDENT.CNO_COUNT || ' ');
13    DBMS_OUTPUT.PUT_LINE(' 平均成绩: ' || EACH_STUDENT.GRADE_AVG || ' ');
14    DBMS_OUTPUT.PUT_LINE(' ----- ');
15  END LOOP;
16 END PRINTGRADES;
17 /
```

过程已创建。

3.实验完成后，撤消建立的基本表和视图。

```
SQL> DROP TABLE SC;  
表已删除。  
SQL> DROP TABLE STUDENT;  
表已删除。  
SQL> DROP TABLE COURSE;  
表已删除。  
SQL> DROP VIEW MALE;  
视图已删除。
```

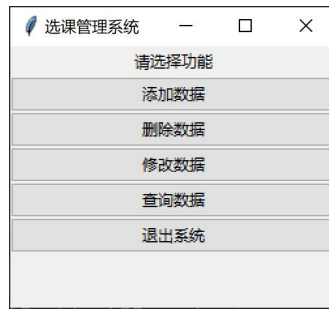
这里和之前一样需要注意的是，由于 Student.Sno 和 Course.Cno 分别是 SC.Sno 和 SC.Cno 的外码，所以需要先删除 SC 表才能再删除 Student 表和 Course 表。

实验六：综合实验：实现一个小型管理信息系统

熟练掌握 Visual C++、Pro*C、Java 或 Python 访问数据库的方法，设计和实现学生通讯录或学生选课的一个小型管理信息系统。要求具有数据的增加、删除、修改和查询的基本功能，并尽可能提供较多的查询功能，用户界面要友好。课程结束前提交上机报告和程序。

在本实验中，基于 Python 程序设计语言和 MySQL 数据库，实现了一个学生选课管理系统。该系统基于 SC 表进行功能实现。该项目除了实现了使用 Python 访问数据库，具有添加、删除、修改、查询的基本功能，还设计了友好的用户交互界面，实现了前后端有效连接。下面具体分析该系统的设计与功能。

首先，程序启动后，会产生一个选择功能的主菜单界面，用户可以选择自己想要使用的功能或者退出系统。可选择的功能有：添加信息、删除信息、修改信息、查询信息。



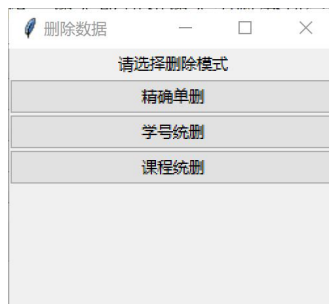
添加数据:

在该页面，用户需要填写待添加的选课信息，包括学号、课程号、成绩、学期和教学班，然后点击确定添加即可。然后系统会调用 MySQL 的 ASERT 指令，对数据库进行插入操作，插入成功和插入失败分别会弹出不同的提示窗口。

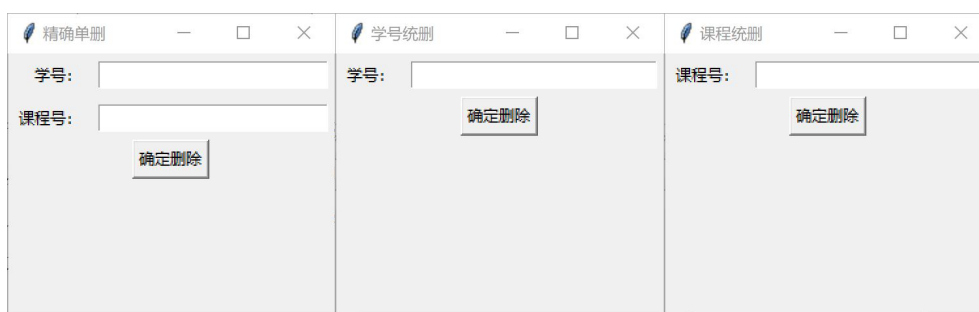


删除数据:

系统提供了 4 种删除数据的方法：包括精确单删、学号统删、课程统删。



精确单删需要用户输入学号和课程号信息，由于 SC 数据库使用了这两个属性构成主码，因此可以唯一确定某个选课信息，精准删除单条记录。学号统删和课程统删则是分别键入学号或者课程号，然后删除该学号或者课程号对应的全部选课信息。三种删除功能的输入界面分别如下图所示。

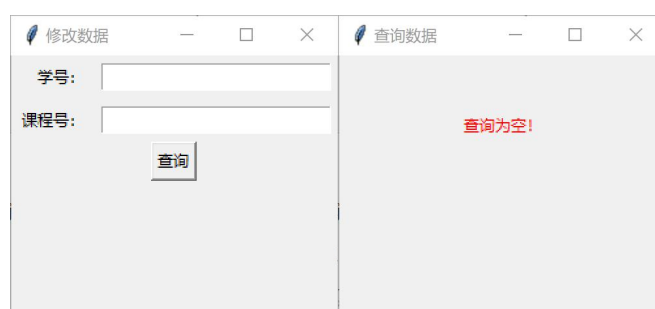


在系统执行删除的过程中，会先使用 SQL 的 SELECT 查询语句，筛选符合条件的记录。如果查询结果数为 0，则会弹出删除失败的提示窗口，如果因为其他系统原因导致 SQL 无法执行删除同样会弹出删除失败窗口。如果系统调用 DELETE 指令能够正确删除对应记录，就会弹出删除成功的提示窗口。



修改数据：

进入修改数据功能后，系统会首先提示输入学号和课程号，通过主码的特性精确 SELECT 精确定位到待修改数据。如果无法找到该记录则会提示查询为空。



如果能够正确查询到数据，则会进入具体修改界面，此时固定学号和课程号信息，需要用户准确输入修改后的数据。然后系统调用 UPDATE 的 SQL 指令对数据库进行修改。正确修改后系统会弹出修改成功提示窗口。如果用户修改不符合表的某些约束，导致修改失败，系统就会弹出修改失败提示窗口。

修改数据

— □ ×

学号: 20180001

课程号: 81001

成绩:

学期:

教学班:

确定修改

修改数据

— □ ×

修改成功!

修改数据

— □ ×

修改失败!

查询数据：

和删除数据类似的，系统提供了精确查询、学号查询、课程查询三种查询模式。

查询数据

— □ ×

请选择查询模式

精确查询

学号查询

课程查询

精确查询需要用户输入学号和课程号，系统调用 SELECT 的 SQL 指令，通过主码的特性实现对唯一记录的查询。学号查询和课程查询分别需要用户提供学号或者课程号，然后系统筛选出符合该条件的全部数据信息。

查询数据

— □ ×

学号:

课程号:

查询

查询数据

— □ ×

学号:

查询

查询数据

— □ ×

课程号:

查询

以学号 20180001 和课程号 81001 为例，展示不同查询模式的反馈结果。

查询结果

— □ ×

学号: 20180001

课程号: 81001

成绩: 85

学期: 20192

教学班: 81001-01

确定

查询结果

— □ ×

学号	课程号	成绩	学期	教学班
20180001	81001	85	20192	81001-01
20180001	81002	96	20201	81002-01
20180001	81003	87	20202	81003-01

确定

查询结果

— □ ×

学号	课程号	成绩	学期	教学班
20180002	81001	80	20192	81001-02
20180003	81001	81	20192	81001-01
20180004	81001	56	20192	81001-02

确定

为了提供友好的使用体验，在查询失败时同样会弹出查询为空的提示窗口。



以上就是选课管理系统的全部功能和实现原理的简要介绍。由于刚刚学习制作前端的 GUI，所以整体看上去比较简陋。某些功能其实并不符合前端制作的要求，例如页面的跳转应该是通过刷新来完成，而非简单的关闭和重开窗口。此外，由于期末月考试较多，未能实现更为复杂多样的用户功能，比如更多设置条件的查询等。事实上，通过该实验，已经初步掌握了使用 Python 对数据库进行控制的方法，更加丰富的功能即通过增加 SQL 指令条件、使用更多的 if 与 else 逻辑判断、更完善更高效的类的架构来实现。未来会基于在本节课上学到的知识，继续完善该系统，并结合人工智能专业的自然语言处理、计算机视觉等知识，添加一些更为高级的功能，如调用大模型 API 的数据库智能交互、数据库输入内容联想及功能建议、成绩单 OCR 扫描实现自动输入数据库等，从而进一步完善在本门课程上学习到的数据库知识。

致谢：

最后，感谢《数据库原理》这门课程，与其他专业课如信息检索与数据挖掘、自然语言处理等形成了良好的知识关联，完善了我们的计算机专业技能。

非常感谢毛宇光老师和陈紫老师对数据库课程对我们的悉心指导，给本课程带来了更多的学习动力与课程魅力。向老师们献上最诚挚的感谢与祝福！

附录：

1.三个基本表样式参考：

学生选课数据库 3 个关系的示例数据

(第 2 章图 2.1)

Student				
学号	姓名	性别	出生日期	主修专业
Sno	Sname	Ssex	Sbirthdate	Smajor
20180001	李勇	男	2000-3-8	信息安全
20180002	刘晨	女	1999-9-1	计算机科学与技术
20180003	王敏	女	2001-8-1	计算机科学与技术
20180004	张立	男	2000-1-8	计算机科学与技术
20180205	陈新奇	男	2001-11-1	信息管理与信息系统
20180306	赵明	男	2000-6-12	数据科学与大数据技术
20180307	王佳佳	女	2001-12-7	数据科学与大数据技术

(a)

Course			
课程号	课程名	学分	先修课
Cno	Cname	Ccredit	Cpno
81001	程序设计基础与 C 语言	4	
81002	数据结构	4	81001
81003	数据库系统概论	4	81002
81004	信息系统概论	4	81003
81005	操作系统	4	81001
81006	Python 语言	3	81002
81007	离散数学	4	
81008	大数据技术概论	4	81003

(b)

SC				
学号	课程号	成绩	选课学期	教学班
Sno	Cno	Grade	Semester	Teachingclass
20180001	81001	85	20192	81001-01
20180001	81002	96	20201	81002-01
20180001	81003	87	20202	81003-01
20180002	81001	80	20192	81001-02
20180002	81002	98	20201	81002-01
20180002	81003	71	20202	81003-02
20180003	81001	81	20192	81001-01
20180003	81002	76	20201	81002-02
20180004	81001	56	20192	81001-02
20180004	81003	97	20201	81002-02
20180205	81003	68	20202	81003-01

(c)

2.实验一至实验五的 SQL 指令见附件 run.sql 脚本文件。

实验六的 Python 代码见附件 main.py 脚本文件。

